

Automatic Detection of Online Abuse and Analysis of Problematic Users in Wikipedia

Charu Rawat, Arnab Sarkar, Sameer Singh, Rafael Alvarado, and Lane Rasberry
University of Virginia, cr4zy, as3uj, ss8gc, rca2t, lr2ua@virginia.edu

Abstract - Today's digital landscape is characterized by the pervasive presence of online communities. One of the persistent challenges to the ideal of free-flowing discourse in these communities has been online abuse. Wikipedia is a case in point, as its large community of contributors have experienced the perils of online abuse ranging from hateful speech to personal attacks to spam. Currently, Wikipedia has a human-driven process in place to identify online abuse. In this paper, we propose a framework to understand and detect such abuse in the English Wikipedia community. We analyze the publicly available data sources provided by Wikipedia. We discover that Wikipedia's XML dumps require extensive computing power to be used for temporal textual analysis, and, as an alternative, we propose a web scraping methodology to extract user-level data and perform extensive exploratory data analysis to understand the characteristics of users who have been blocked for abusive behavior in the past. With these data, we develop an abuse detection model that leverages Natural Language Processing techniques, such as character and word n-grams, sentiment analysis and topic modeling, and generates features that are used as inputs in a model based on machine learning algorithms to predict abusive behavior. Our best abuse detection model, using XGBoost Classifier, gives us an AUC of ~84%.

Index Terms - Machine Learning, Natural Language Processing, Online Abuse, Text Mining

INTRODUCTION

In today's digital world, where information is readily available and everyone has the ability to easily connect with each other, there are immense possibilities for knowledge sharing and community building. At the same time, there is a dark side to this freedom. According to the Pew Research Survey of 2017, ~41% of Americans have experienced personal attacks online and ~66% have observed attacks directed towards others [1]. Wikimedia has not remained untouched by this phenomenon. In the last few years, the community has seen a steady increase in both the number of incidents and variety of attacks (Wiki-hounding, sock puppetry, user talk harassment, posting personal information, etc.) [2].

Wikimedia has realized that online harassment of its editors can have a detrimental effect on the growth of its platform, and so the community has taken proactive steps to

create awareness around such issues and has put into place, a well-defined and structured "no personal attack" policy for all of its members [3]. Currently, when users misbehave there is a process by means of which they might get blocked after human evaluation [4]. While human evaluation works in some ways, it is not a solution that scales well with the growth of Wikimedia projects. Cases sometimes fall through the cracks of the process of human scrutiny. In 2017, the Google Ex: Machina study [5] quantified the impact of this effect on the English Wikipedia - one-fifth of the personal attacks get reported, which suggests that majority of such attacks go unreported in the community.

The contributions of this paper are threefold. Firstly, we explore different sources to gather user-level data from Wikipedia. In the data acquisition section, we discuss the different data sources and approaches used to acquire the data. Secondly, we explore the characteristics of the different blocked users by their block types and compare their behavioral features with non-blocked users. Thirdly, we create a machine learning based abuse detection model. We generate features by leveraging natural language processing techniques and deriving features from the user texts, which are then fed into different machine learning models. We compared the performance across the different models and checked the robustness of our model by training them on Google Ex: Machina dataset [5] as well.

RELATED WORK

In the area of automatic abuse detection, a variety of work has been done by many researchers. In the area of online toxicity detection, Justin et. al. [6], looked at the writing styles and post attributes across two sets of users- Future Banned Users (FBU) and Never Banned Users (NBU), spanning over 3 different user communities. The analysis showed that the writing actually worsens for the FBU group after they get blocked which is attributed to overly harsh community feedback against these users.

In another study by David Noever [7], features of the raw text such as syntax, sentiment, emotions, etc. were used to generate 62 classifiers using 19 different algorithms. Based on accuracy measures and relative execution times, it was noted that tree-based algorithms provide the best feature importance for detecting toxicity. Our research takes these findings into account by going beyond just analyzing the use of offensive words in user comments.

Considerable research has been done with a focus on studying these dynamics in the English Wikipedia

community as well. A Wiki Trust Model (WTM) [8] was developed by Sara et. al. which assigns a reputation score to each user depending on their edit contributions. The model looked at the text of a new revision added to an article versus the text of the previous revision on that article. Users whose content remains on the wiki get good scores; conversely, users whose contribution is deleted get penalized. In the same vein, Sara et. al. also developed a vandalism detection model [9], capturing 66 features from both edit textual features and edit meta-features and then optimizing these features using Lasso optimization to develop a classification model. However, this research is primarily focused on vandal detection over the article edits and not on abusive user behavior.

In the area of abuse detection in conversations between users in the English Wikipedia, some of the most relevant work has been done in the Google Ex: Machina study by Ellery et. al. [5], where they extracted and analyzed large corpus of the English Wikipedia data utilizing human annotation and a robust machine learning model for abuse detection. Even though our classification problem is same as theirs, the methodology is different as they had annotations for each comment hence, they detected abuse at the comment level, whereas we classify users as a block or not based on a collection of their recent comments.

All revisions or edits in English Wikipedia happen over 35 partitioned areas called “Namespaces”. For our paper, we developed a web scraping methodology to obtain the user comments from the relevant namespaces in the English Wikipedia whereas Ellery et. al. extract data using XML dumps that are generated by Wikipedia. The reasons for using a different methodology in data extraction is elaborated below in the data acquisition section. We also observe that other papers [5]-[7]-[8] have focused on abuse detection using just the text data, but we wanted to incorporate textual and behavioral features of the users to make user level predictions regarding the propensity of blocks in the future. Keeping that in mind, we have leveraged other data sources in conjunction with the text data to gain a better understanding of how and why users are blocked in the English Wikipedia community.

DATA ACQUISITION

All the data used in our work has been made publicly available by Wikipedia. One major challenge to working with these data is the lack of a unified dataset that can be used to query multiple, cross-cutting aspects of users contributions. As a result, a considerable amount of time and resource was spent on data acquisition and preparation. Wikipedia makes its data available in a variety of ways. Below is a summary of the data we used, along with its purpose, the data source and the method used in acquiring it.

I. Ipblocks table and Revision table data

To access information regarding all users who have been blocked in the English Wikipedia because of misconduct or abusive behavior, we accessed data stored in the ipblocks

table [10]. This data provides us with a record of all users who have been blocked between Feb 2004 - Nov 2018 in the English Wikipedia, which amounts to 1,172,642 rows. There are a total of 20 attributes in this table that give information about the blocked users. We have used 8 of these attributes in our analysis (Table I).

TABLE I
IPBLOCKS TABLE ATTRIBUTES

Attributes	Description
ipb_id	primary key
ipb_address	blocked ip address in dotted-quad form or username
ipb_user	blocked user_id or 0 for ip blocks
ipb_by	user_id of the admin who made the block
ipb_by_text	text username of the admin who made the block
ipb_timestamp	creation (or refresh) date in standard ymdhms form
ipb_expiry	expiry time set by the admin at the time of the block. A standard timestamp or the string ‘infinity’
ipb_reason	reason for the block given by the admin

The revision table [10] holds metadata for every edit done to a page across all the namespaces in Wikipedia. Every edit of a page creates a revision row, which holds information about the revision edit (Table II). The data in this table was aggregated and transformed to the user level to gain insights into the revision activity patterns of a user. The revision data pulled spans Jan 2017 - Aug 2018 which resulted in 95,761,694 rows for 6,126,510 users.

Both ipblocks and revision are MariaDB tables on the enwiki schema [10] of Wikimedia’s backend database servers, which can be accessed via Wikimedia’s Toolforge [11] hosting environment. We performed SSH tunneling between the enwiki schema on Toolforge and our AWS Sagemaker instance. We then made use of a combination of SQL queries and Unix shell SCP commands to get the appropriate amount of data as required from the two tables. The data was dumped as tab-separated text files on a specified location in AWS Sagemaker instance.

TABLE II
REVISION TABLE ATTRIBUTES

Attributes	Description
rev_id	primary key for each revision
rev_user	user_id of the user who made this edit. Value is 0 for anonymous users
rev_user_text	text of the editor’s username or the IP address of the editor if the revision was done by an unregistered user
rev_timestamp	timestamp of the edit
rev_minor_edit	binary value that records whether the user marked the “minor edit” checkbox
rev_deleted	it’s a bitfield in which values are populated for deletion. 0 in case of no deletion
rev_len	the length of the article after the revision in bytes

II. User comment data

For the abuse detection model, we looked into the comments between users on the English Wikipedia. We focus our text-based modeling on talk namespaces as we are concerned with detecting abuse in comments between users and not their edits per se. Within talk namespaces, we specifically focus on “User Talk” and “Article Talk”, as these two contain the maximum amount of user comments compared to others.

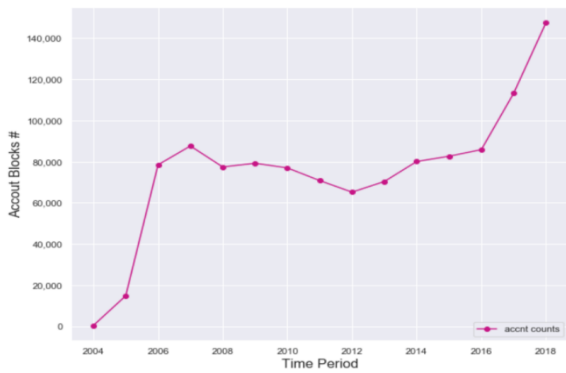


FIGURE I
ACCOUNT BLOCKS OVER THE YEARS

This user comment data is not stored in any of the structured tables in the enwiki database schema [10]. There are two ways to access this data, both detailed below -

- **XML Dumps:** We used the pages-meta-history files from the Nov 2018 released XML dumps, to extract user level comments across the different pages. [13]. We built an auto-downloader web scraper [12] to download these files from the Wikipedia mirror sites [13] on to our AWS Sagemaker instance. Each of these files was decompressed and processed through an XML parser built using the MediaWiki utilities [14] on Python. A problem with these XML files is that each text (characterized by a single rev_id) in page_id is not unique to a user and contains text from the previous rev_id as well. Due to this, it is necessary to compute diffs between texts pertaining to sequential rev_id's while parsing the XML files. The data in these XML files are stored at a page_id level, which limits the availability of the historical information of each user. One would have to potentially parse through 550+ compressed XML files that total 2TB+ to get all historical edits of any user. Such a process can be computationally exhaustive and expensive.
- **Web scraping:** The layout of the English Wikipedia namespace is such that each conversation thread is a separate page, and each comment from a user becomes a diff on that conversation page. We built a wiki-user-text web scraper [12] to scrape the text data on these diffs for the required users. The text data is dumped in CSV format on an AWS Sagemaker instance. We scrape a maximum of 20 comments from both the namespaces. The entire text corpus consists of 503,355 comments for 50,646 users. This approach gives us all the required data i.e. each comment unique to a user. The efficiency of this process can be improved by using distributed computing.

III. Annotated data from the Google Ex: Machina study

Google's Ex Machina study [5] carried out human annotation exercise for each user comment. This data is readily available on Figshare [15]. The comments were graded on three different criterion – personal attacks, aggressiveness, and toxicity, by 10 annotators. We averaged scores from all

annotators to create a single label. The total data consists of 197,578 rows where each row is a unique user comment. This data was used to validate the results from the abuse detection model.

EXPLORATORY DATA ANALYSIS

Users can contribute to Wikipedia using their registered credentials such as their username or contribute anonymously using their IP addresses. A "user" refers to a registered account or an IP address and it does not refer to a real-world individual. We analyzed the block data to study block trends over the years and present a statistical comparison of blocked and non-blocked users' behavior using the revision activity data. The findings from this analysis laid a strong foundation for our work on modeling abuse detection.

I. Block Data

As of Oct 2018, 1.02 million unique users have been blocked in the English Wikipedia, out of which 91.5% are registered users and 9.5% are anonymous users. We visualized the blocked user accounts on a time series scale, which gave us an insight into how trends have evolved over the years. Our findings suggested that the number of blocked users rose in 2017 and 2018 (Figure I), but the increase in blocked accounts was significant in Sep, Oct 2018. We noted this rise in blocked users was primarily due to anonymous users getting blocked. To examine this trend further, we looked into other attributes related to blocked users (Table I).

A user can be blocked for a variety of reasons by the admin. The inconsistency in the way the block reason field is populated by admins, makes it challenging to use it its raw format. We relied on using regular expressions to extract key block reasons such as "spam", "vandalism", etc. from ipb_reason (Table I). These keywords are based on the Wikipedia block template which outlines some of the most common reasons for which users can be blocked [16]. On visualizing the share of reasons for which users are blocked, we observed that this share has been evolving over time. We noticed an increase in the share of reason "proxy" [16] in late 2018 which seemed in line with the spike in blocked user accounts in 2018. Tying these insights together, we could conclude that in Sep, Oct 2018, there was a significant rise in anonymous users getting blocked for being proxies. The findings from our analysis also suggest that registered users are more likely to get blocked for reasons such as vandalism, spam, sock puppetry [16] whereas anonymous users are more likely to get blocked for reasons such as proxy, webhost, and school blocks [16]. A reasonable share of users who get blocked often go untagged as well and we categorized them as not available under the block reason field in our analysis.

On analyzing the ipb_by_text attribute (Table I), our analysis showed that in 2018, 5 admins accounted for enacting 50% of the blocks. 30% of the total blocks enacted in 2018 were made by 1 admin, who mostly blocked anonymous users for being proxies. We could conclude from our analysis that this dynamic led to the rise in blocked users in late 2018.

II. Revision Activity Data

We analyzed the revision activity data to gain insights into how users behaved in the weeks leading up to them getting blocked. We also explored if the reasons for them getting blocked can be tied to their temporal activity patterns. We aggregated features such as `rev_count`, `rev_length`, `minor_rev`, `deleted_rev` (Table II) on a weekly basis over the most recent 8-week window available for every user. We looked at the 8 weeks leading up to the block for users and observed that 92.3% of the blocked users are highly active in the week prior to them being blocked. Our findings also suggest that users blocked for reasons such as vandalism, proxy, spam [16] tend to make anywhere between 5-15 revisions on average every week within their recent 8-week window.

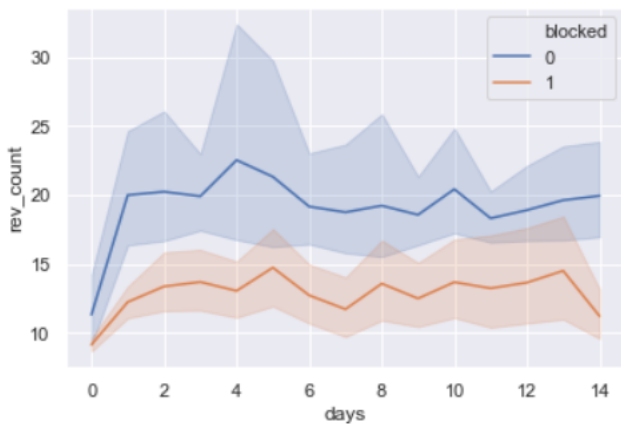


FIGURE II
REVISION COUNTS FOR BLOCKED VS NONBLOCKED USERS

The reasons for which users get blocked can potentially be dependent on the length of their revision edits as well. Users who get blocked as proxies tend to have a varying spread of weekly average revision length in their 8-week window as compared to users who are blocked for other reasons.

To analyze how activity differs across blocked and non-blocked users, we looked at the revision activity data aggregated on a daily basis over the most recent 2-week window available of a user. We found that non-blocked users generally make a higher number of edits on a daily basis compared to blocked users (Figure II). The proportion of minor edits made by non-blocked users is double the number of minor edits made by blocked users. We also noted that in the 2 weeks leading to a user getting blocked, there is an increase in the proportion of deleted edits for such users that is likely a result of their early disruptive behavior on the platform.

The exploratory data analysis on the revision activity data highlighted the difference in editing patterns of users who get blocked vs users who don't. This analysis also points us towards areas where the scope of such a study can be expanded as listed in the conclusions section.

We created a toxicity scoring system and computed scores for each user comment. This score is the probability value generated from the model with 0 indicating the comment is non-toxic to 1 denoting an extremely toxic comment. We used different natural language processing techniques along with multiple machine learning algorithms to solve this task.

I. Corpus

We used the data scraped and processed using wiki-user-text scraper as mentioned earlier in subsection-II of Data Acquisition. For each user, we have their recent 40 comments within the time frame Jan 2017-Aug 2018.

II. Data Pre-Processing

The raw data scraped from Wikipedia consisted of plenty of irrelevant words and symbols, due to the syntactic rules of Wikipedia markup and formatting. In some cases, comments would contain the user's username (for example `[[User:username|username]]`). The time and date were also added in each comment. We used regular expression extensively to ensure that all irrelevant information was removed from the corpus.

III. Annotation

For our analysis, we have a list of blocked users, as mentioned earlier in subsection I of Data Acquisition. Unfortunately, in the blocking system of the English Wikipedia, there is no data recorded that points to the particular comment(s) which lead to a user getting blocked. The amount of time between a user posting an abusive comment to the user being blocked varies. For some users, the block might be instantaneous whereas for others the block might take multiple days to be enforced. To determine the number of comments that should be used in the abuse detection model, we adopt a heuristic approach relying upon our findings from the exploratory data analysis. The approach consists of two main steps. As a first step, we considered comments made by users in their latest week since one of our findings suggested that 92.3% of the blocked users are majorly active in the week prior to their block. As a second step, we analyzed the average number of comments made by blocked users and chose to only consider the five most recent comments made in that 1-week period. Even after these measures, it is difficult to confidently conclude that all the comments are abusive. In order to work around this issue, we created a new aggregated corpus using the pre-processed comments of each user. This ensured that for blocked users, the aggregated corpus has a high probability of containing abusive comments.

IV. Feature Engineering

- Text derived features: We derived multiple features based on the text itself. Key features are captured in the table III.

TABLE III
TEXT DERIVED FEATURES

Feature name	Description
num_digit	Numeric digits divided by total number of characters
cap_char	Capital characters divided by total number of characters
cap_non_cap	Ratio of capital to non-capital characters
nword	Count of number of words
spec_char	Special characters divided by number of characters
unq_word	Unique words divided by total number of words
avg_char	Average length of characters in a word

- **Sentiment Analysis:** We used NLTK based Vader sentiment analyzer, as it is fine-tuned for social media texts. We applied sentiment analysis on the text before lemmatizing, and lower casing it.
- **Word n-gram:** We derived the word level n-grams, after lemmatizing and removing common English stop words. We then mapped every document to a feature vector, using term frequency-inverse document frequency encoding (TF-IDF). We had two hyperparameters, the number of n-grams, and the number of features. We used grid search to select values for them, and finalized on 1,500 top features, with 1-2 word n-gram.
- **Character n-gram:** We also derived character level n-grams, as in social texts, spelling mistakes and the use of special characters to mask abusive words is very prevalent. Similar to word n-gram, we created features using TF-IDF and performed grid search to finalize on the hyperparameters. We finalized on 5,000 top features, with 2-6 char n-grams.
- **Topic modeling:** We leveraged topic modeling to derive additional features from our text corpus. To decide on the number of topics we performed grid search and finalized on 15 topics.
- **Username based features:** We derived character level n-grams for usernames, as we observed user names often contain abusive content as well. Then we created features using TF-IDF and performed grid search to finalize on 300 top features, with 1-2 char n-grams.

V. Implementing machine learning algorithms

We combined all the different features mentioned above (except username-based features) to create a consolidated dataset and then applied multiple machine learning algorithms, which are useful for classification, such as Logistic Regression, SVM, Random Forest, Gradient Boosted Trees, and XGBoost. We used 75%-25% split between train and test data and used k-fold cross-validation to measure the AUC scores to compare the different models.

VI. Model comparison with Google Ex: Machina data

To compare the performance of our best performing model, we trained the model (recreating all the same features) based on Google Ex Machina dataset [5]. For each comment the model predicted a toxicity score and then, we calculated the average toxicity score of each user. We used this value to

predict a binary label which indicates whether a user is blocked or not.

VII. Final model

We added the username-based features to the best performing model. To classify whether a user is blocked or not, we determined the optimal probability threshold value by performing grid search and analyzing statistics such as AUC, F1-score, Accuracy across different threshold values.

RESULTS

Table IV captures the drop in the number of unique users after we perform the pre-processing steps on the dataset.

TABLE IV
UNIQUE USER COUNTS

Data	Blocked Users	Non-Blocked Users
Initial Corpus	19,580	31,066
After data pre-processing	12,186	25,748

I. Model selection

To compare the different machine learning techniques for abuse detection, we used k-fold cross-validation and analyzed the AUC scores (Table V).

TABLE V
MODEL CV AUC SCORES

Model	AUC
Logistic Regression	65.36 %
Linear SVM	66.03 %
Random Forest Classifier	63.54 %
Gradient Boosting Classifier	66.74 %
XGBoost Classifier	68.26 %
XGBoost Classifier (with username features)	83.10 %

II. Model comparison

The XGBoost Classifier model trained on the Google Ex: Machina dataset [5] gives ~71% accuracy compared to the ~75% accuracy obtained from the same classifier trained on the aggregated user comment corpus.

III. Model Threshold finalization

Table VI captures the different metrics. We observed that a threshold of 0.4 seemed most balanced, with good accuracy, AUC and recall, and the highest F1-score.

TABLE VI
PERFORMANCE METRICS ACROSS VARIOUS THRESHOLD VALUES

Threshold	AUC	Accuracy	F1-Score	Precision	Recall
0.3	84.9 %	85.1 %	85.0 %	73.0 %	84.0 %
0.4	84.0 %	85.7 %	86.0 %	77.0 %	79.0 %
0.5	83.1 %	86.2 %	86.0 %	81.0 %	74.0 %
0.6	81.1 %	85.8 %	85.0 %	85.0 %	68.0 %
0.7	78.3 %	84.7 %	84.0 %	88.0 %	60.0 %

IV. Toxicity score comparison

We analyzed the recent six toxicity scores for blocked and non-blocked users. As shown in Figure III, the blocked users'

scores are skewed towards the right depicting high toxicity, whereas they are skewed towards left for non-blocked users.

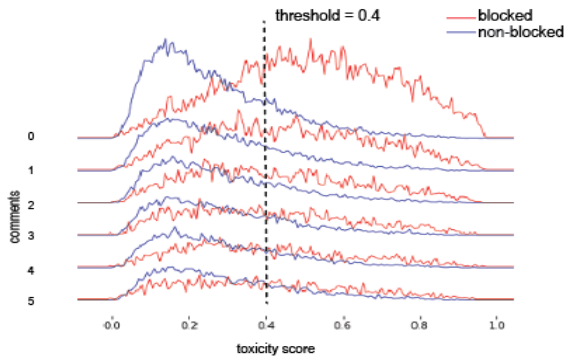


FIGURE III
TOXICITY SCORE EVOLUTION FOR USERS

CONCLUSION

Through this paper, we were able to develop a framework for a data-driven approach to detect abuse in the English Wikipedia community. A large part of this framework was understanding the data sources and the methods that could be leveraged in acquiring that data. The first objective achieved through this paper was to successfully develop two methods to acquire the user comment data through different sources. In this paper, we successfully outlined the different reasons for the suitability of each data source and the data acquisition process. Both the processes – the wiki-comment web scraper and the XML dump extractor, can be leveraged across other namespaces and other Wikipedia communities as well.

The second objective achieved through our paper was in discovering some of the interesting dynamics within the blocking ecosystem of the English Wikipedia by analyzing the block data and the revision activity data. We uncovered a change in the blocking trends of the English Wikipedia that took place in 2018 and were able to provide a rationale behind the change. The exploratory data analysis on the revision activity data highlighted how revision activity patterns differ across various groups of users especially when we compare the patterns of blocked users vs non-blocked users. Our takeaways from this analysis make a strong case for utilizing this data further for the early detection of problematic users.

Our third objective was to build a model to detect blocked users. We were able to successfully develop and evaluate different abuse detection models trained on the text corpus generated from the user comment data. These models were implemented using various machine learning algorithms like Linear SVM, Logistic Regression, Random Forests, Gradient Boosting, and XGBoost, and we found that the best performance was given by the XGBoost Classifier with an AUC of 84%.

In the future, we plan to implement a user risk prediction model which would be able to predict and flag users who are at risk of getting blocked in the future. This model will make use of the “toxicity scores” generated by the abuse detection model. In addition to these “toxicity scores”, the proposed

model will also leverage the temporal user activity features generated out of the revision activity data. Word embedding, deep neural network like LSTMs can further be implemented to improve the accuracy of our models.

ACKNOWLEDGMENT

The authors would like to thank Patrick Earley and Claudia Lo from the Trust & Safety team at the Wikimedia Foundation for their encouragement and helpful insights during the course of this project.

REFERENCES

- [1] M. Duggan., “Online harassment. Pew Research Center, 2017”. Available: <http://www.pewinternet.org/2017/07/11/online-harassment-2017/>.
- [2] “Wikipedia. Wikimedia: Wikimedia Harassment”. Available: <https://en.wikipedia.org/wiki/Wikipedia:Harassment>.
- [3] “Wikipedia. Wikipedia: No personal attacks”. Available: https://en.wikipedia.org/wiki/Wikipedia:No_personal_attacks
- [4] “Wikipedia: Blocking policy”. Available: https://en.wikipedia.org/wiki/Wikipedia:Blocking_policy
- [5] Ellery Wulczyn, Nithum Thain, Lucas Dixon, “Ex Machina: Personal Attacks Seen at Scale”, 2017 International World Wide Web Conference Committee (IW3C2), published under Creative Commons CC BY 4.0 License. WWW 2017, April 3–7, 2017, Perth, Australia.
- [6] Justin Cheng, Cristian Danescu-Niculescu-Mizil, Jure Leskovec, “Antisocial Behavior in Online Discussion Communities”, Proceedings of the Ninth International AAAI Conference on Web and Social Media.
- [7] David Noever, “Machine Learning Suites for Online Toxicity Detection”. Available: <https://arxiv.org/ftp/arxiv/papers/1810/1810.01869.pdf>
- [8] Sara Javanmardi, Yasser Ganjisaffar, Cristina Lopes and Pierre Baldi, “User Contribution and Trust in Wikipedia”, 5th International Conference on Collaborative Computing: Networking, Applications, and Worksharing, 2009
- [9] Sara Javanmardi, David W. McDonald, Cristina V. Lopes, “Vandalism Detection in Wikipedia: A High-Performing, Feature-Rich Model and its Reduction Through Lasso”, WikiSym’11 October 3–5, 2011, Mountain View, CA, USA.
- [10] “Wikipedia Commons. MediaWiki_1.28.0_database_schema.svg”. Available: https://upload.wikimedia.org/wikipedia/commons/9/94/MediaWiki_1.28.0_database_schema.svg
- [11] “Wikipedia Wikitech. Portal: Toolforge”. Available: <https://wikitech.wikimedia.org/wiki/Portal:Toolforge>
- [12] “Github. UVA-DSI-2019-Capstones/TST/data_acquisition”. Available: https://github.com/UVA-DSI-2019-Capstones/TST/tree/master/data_acquisition
- [13] “Wikipedia: Data Dumps”. Available: https://meta.wikimedia.org/wiki/Data_dumps
- [14] “Mediawiki Utilities”. Available: <https://pythonhosted.org/mediawiki-utilities/>.
- [15] “Figshare. Wikipedia Talk”. Available: https://figshare.com/projects/Wikipedia_Talk/16731
- [16] “Wikipedia. Category: User Block Templates”. Available: https://en.wikipedia.org/wiki/Category:User_block_templates.